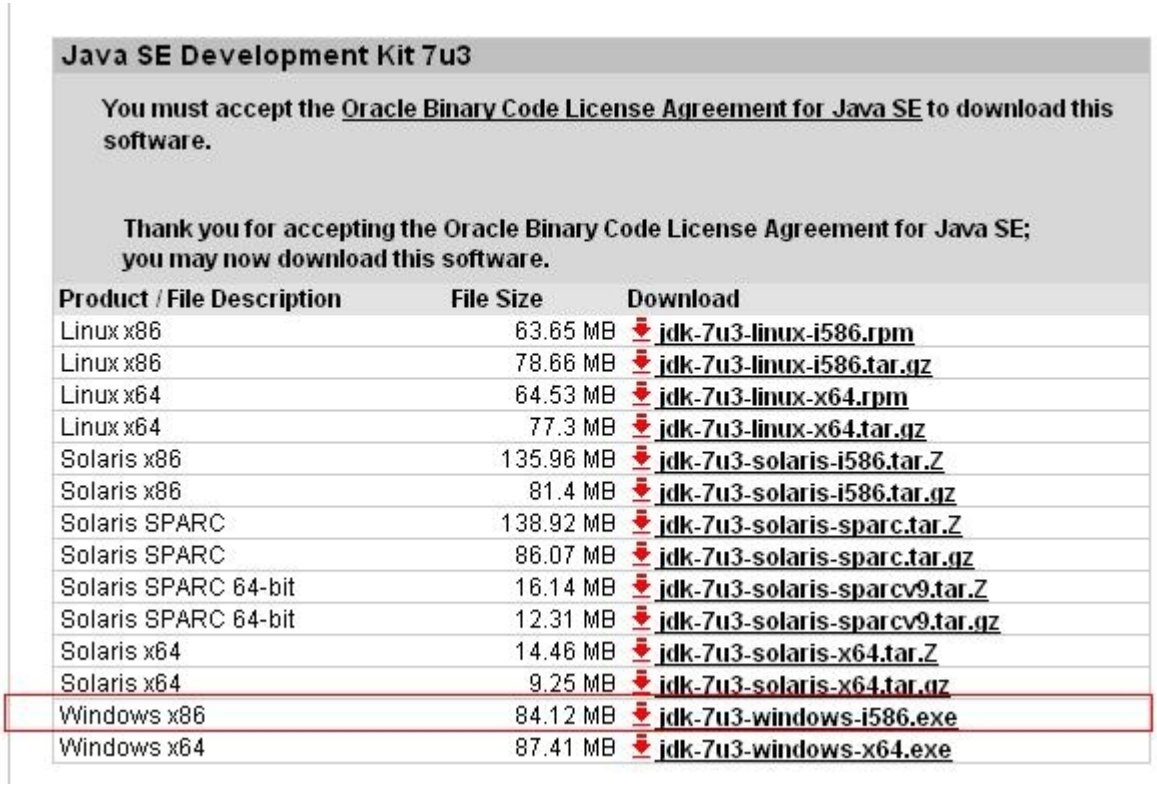
Lazarus 开发 Android 应用程序指南(中文首发)

本文件描述的安装环境为 Windows XP SP3。

1. 下载和安装配置 FPC 和 Lazarus 安卓（Android）开发环境：

1.1 下载安装 JDK：http://www.oracle.com/technetwork/java/javase/downloads/index.html，如图 1 和图 2 所示。





Java SE Development Kit 7u3

You must accept the Oracle Binary Code License Agreement for Java SE to download this software.

Thank you for accepting the Oracle Binary Code License Agreement for Java SE; you may now download this software.

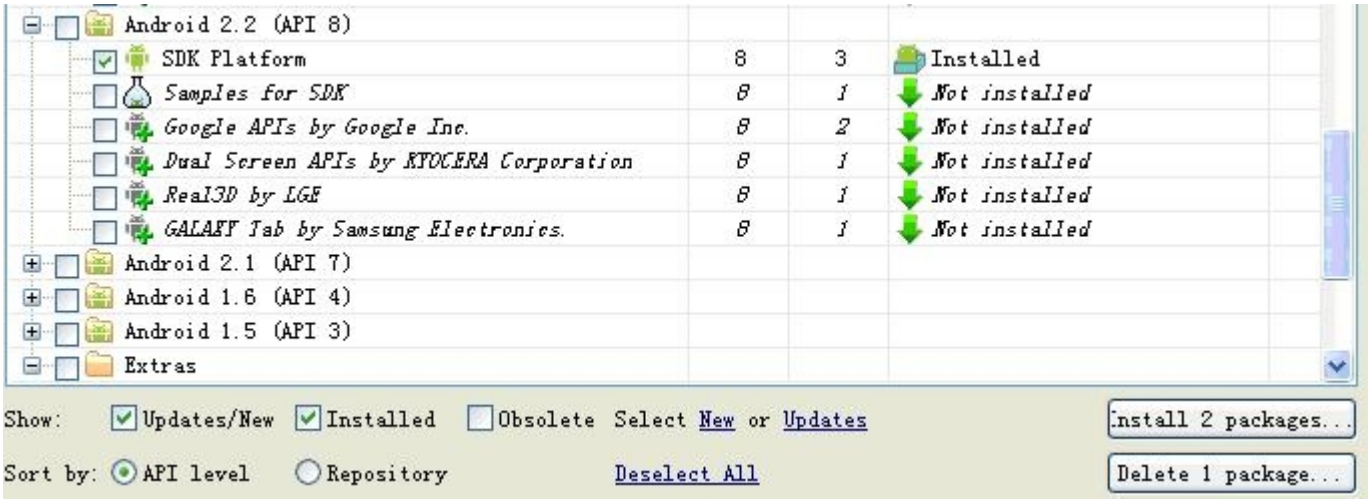
Product / File Description	File Size	Download
Linux x86	63.65 MB	jdk-7u3-linux-i586.rpm
Linux x86	78.66 MB	jdk-7u3-linux-i586.tar.gz
Linux x64	64.53 MB	jdk-7u3-linux-x64.rpm
Linux x64	77.3 MB	jdk-7u3-linux-x64.tar.gz
Solaris x86	135.96 MB	jdk-7u3-solaris-i586.tar.Z
Solaris x86	81.4 MB	jdk-7u3-solaris-i586.tar.gz
Solaris SPARC	138.92 MB	jdk-7u3-solaris-sparc.tar.Z
Solaris SPARC	86.07 MB	jdk-7u3-solaris-sparc.tar.gz
Solaris SPARC 64-bit	16.14 MB	jdk-7u3-solaris-sparcv9.tar.Z
Solaris SPARC 64-bit	12.31 MB	jdk-7u3-solaris-sparcv9.tar.gz
Solaris x64	14.46 MB	jdk-7u3-solaris-x64.tar.Z
Solaris x64	9.25 MB	jdk-7u3-solaris-x64.tar.gz
Windows x86	84.12 MB	jdk-7u3-windows-i586.exe
Windows x64	87.41 MB	jdk-7u3-windows-x64.exe

1.2 下载安装 Android SDK

从以下网址下载安卓 SDK 开发包：http://developer.android.com/sdk/index.html，选择如图 3 所示。

Platform	Package	Size	MD5 Checksum
Windows	android-sdk_r16-windows.zip	29562413 bytes	6b926d0c0a871f1a946e65259984701a
	installer_r16-windows.exe (Recommended)	29561554 bytes	3521dda4904886b05980590f83cf3469
Mac OS X (intel)	android-sdk_r16-macosx.zip	26158334 bytes	d1dc2b6f13eed5e3ce5cf26c4e4c47aa
Linux (i386)	android-sdk_r16-linux.tgz	22048174 bytes	3ba457f731d51da3741c29c8830a4583

首先安装 JDK，然后安装 Android SDK。安装完 SDK 后需要下载相应的 SDK 包，打开程序->Android SDK tools->SDK Manager 安装 SDK 2.2：



后面的例子会用到。

1.3 下载安装 Android NDK ：http://developer.android.com/sdk/ndk/index.html

Platform	Package	Size	MD5 Checksum
Windows	android-ndk-r7b-windows.zip	80346206 bytes	c42b0c9c14428397337421d5e4999380
Mac OS X (intel)	android-ndk-r7b-darwin-x86.tar.bz2	73817184 bytes	6daa82ca6b73bc0614c9997430079c7a
Linux 32/64-bit (x86)	android-ndk-r7b-linux-x86.tar.bz2	64349733 bytes	0eb8af18796cdaa082df8f7c54ad7f9a

图 5

注意：此 NDK 后来改为下载：android-ndk-r7-windows.zip 这个包，上面那个包中的 arm-linux-ld.exe 编译时会出错，使用这个包中的不会出错了。地址：http://dl.google.com/android/ndk/android-ndk-r7-windows.zip 。
NDK 解压缩就可以了。

1.4 检查以下路径的工具是否正确，由 NDK 安装的。例如我的环境：D:\AndroidFPC\NDK\android-ndk-r7b\toolchains\arm-linux-androideabi-4.4.3\prebuilt\windows\bin。检查库路径：D:\AndroidFPC\NDK\android-ndk-r7b\platforms\android-8\arch-arm\usr
备注：已成功把最新 Svn trunk 版 Lazarus0.9.31 和 FPC 2.5.1 打包成一个安装文件。可以正常安装使用了，不会出错，可以正常生成安卓文件。以下 1.5 到 1.10 的步骤都可以跳过。2.1 的出错后重定位部分的步骤也可以跳过了。这个安装包放 115 网盘。网盘地址在第 3 帖最后。

1.5 **下载** FPC 2.5.1（1 月 19 日版），首先你需要有 SVN 下载工具。新建一个文件夹，然后 SVN 检出 FPC 源码，版本号是 16790。新建一个文件夹 D:\AndroidFPC\FPC251 用于存放 SVN 源码，右击文件夹选择 SVN 检出，版本号 16790，如图 6 所示

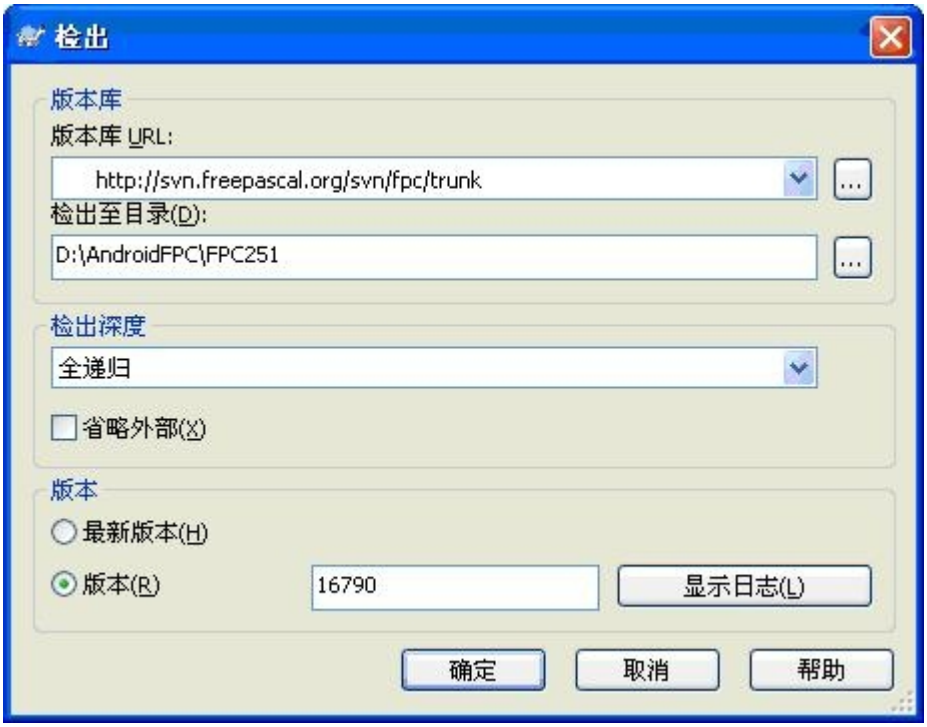


图 6

1.6 **安装** FPC 2.4.4 。简单的安装法是安装 Lazarus 0.9.30.2，因为已经包含了。用来编译生成 ppcrossarm.exe ，后面会介绍。

1.7 **创建构建脚本并且构建它。**先创建输出文件夹：D:\AndroidFPC\FPC251\output
创建一个批处理文件在以下文件夹 ：D:\AndroidFPC\FPC251 ，文件名 build.bat，内容如下：

```
SET PATH=D:\lazarus\fpc\2.4.4\bin\i386-win32\
make crossinstall CPU_TARGET=arm OS_TARGET=linux
CROSSBINDIR=D:\AndroidFPC\NDK\android-ndk-r7b\toolchains\arm-linux-androideabi-4.4.3\
prebuilt\windows\bin OPT=-dFPC_ARMEL BINUTILSPREFIX=arm-linux-androideabi-
INSTALL_PREFIX=D:\AndroidFPC\fpc251\output
pause
```

现在双击运行 build.bat，5 分钟后，在 D:\AndroidFPC\FPC251\output 文件夹中会有两个文件夹：bin 和 units，FPC 编译器在 D:\AndroidFPC\FPC251\output\bin\i386-win32 文件夹中，名为：ppcrossarm.exe ，成功构建，这个文件是重点。

1.8 **复制交叉编译工具。**
从 D:\AndroidFPC\NDK\android-ndk-r7b\toolchains\arm-linux-androideabi-4.4.3\prebuilt\windows\bin 复制 3 个交叉二进制工具到 D:\AndroidFPC\FPC251\output\bin\i386-win32 文件夹中。这三个文件的名称为：arm-linux-androideabi-as.exe，arm-linux-androideabi-ld.exe，arm-linux-androideabi-strip.exe ，重新命名为：arm-linux-as.exe，arm-linux-ld.exe，arm-linux-strip.exe 。这里注意 1.3 的描述。

1.9 **安装新版** Lazarus。
现在我们需要安装最新版本 lazarus，把原来的 Lazarus 0.9.30.2 卸掉，不要忘记同时清理 C:\Documents and Settings\你的用户名\Local Settings\Application Data 中 Lazarus 的文件夹。

从 ftp.freepascal.org/ /pub/lazarus/snapshots/下载最新版本的 Lazarus 安装。我安装的是：Lazarus-0.9.31-35521-fpc-2.7.1-20120221-win32.exe 这个版本。新版已经打包，在 115 网盘中，自己打包的。下载地址请看第 3 帖结尾。

1.10 合并工具到 Lazarus 环境。

复制 D:\AndroidFPC\FPC251\output\bin\i386-win32 中所有文件到 Lazarus 安装文件夹 D:\lazarus\fpc\2.7.1\bin\i386-win32 ，复制文件夹 D:\AndroidFPC\FPC251\output\units\arm-linux 到 D:\lazarus\fpc\2.7.1\units 中。

到了这一步，我们在 Windows 的中的 lazarus 部分就配置成功了。下面还有工具及设置要调整。

1.11 下载安装设置 Ant 。从以下地址：http://www.apache.org/dist/ant/binaries/ ；选择 apache-ant-1.8.2-bin.zip 下载就可以了。下载后解压缩，把解压后的文件夹放在：C:\Program Files 文件中，然后在环境变量中配置如以下图：



图 7



图 8

至此，所有 windows 配置及工具已经全部下载安装完成，可以开始调试程序了。

2．编译 lazarus 中的 Android 示例程序。

示例是在 lazarus/examples/androidlcl/androidlcltest.lpi。我这里把示例程序放在了 D:\AndroidFPC\demo\demo 中，所以以下示例都以此为准。

2.1 重新设置 Lazarus 。

在打开示例程序之前，我们先要把 lazarus 重新设置一下，进入最新安装的 Lazarus 文件夹 lazarus\fpc，把 2.7.1 改为 2.5.1，然后重新打开 lazarus，会出错：

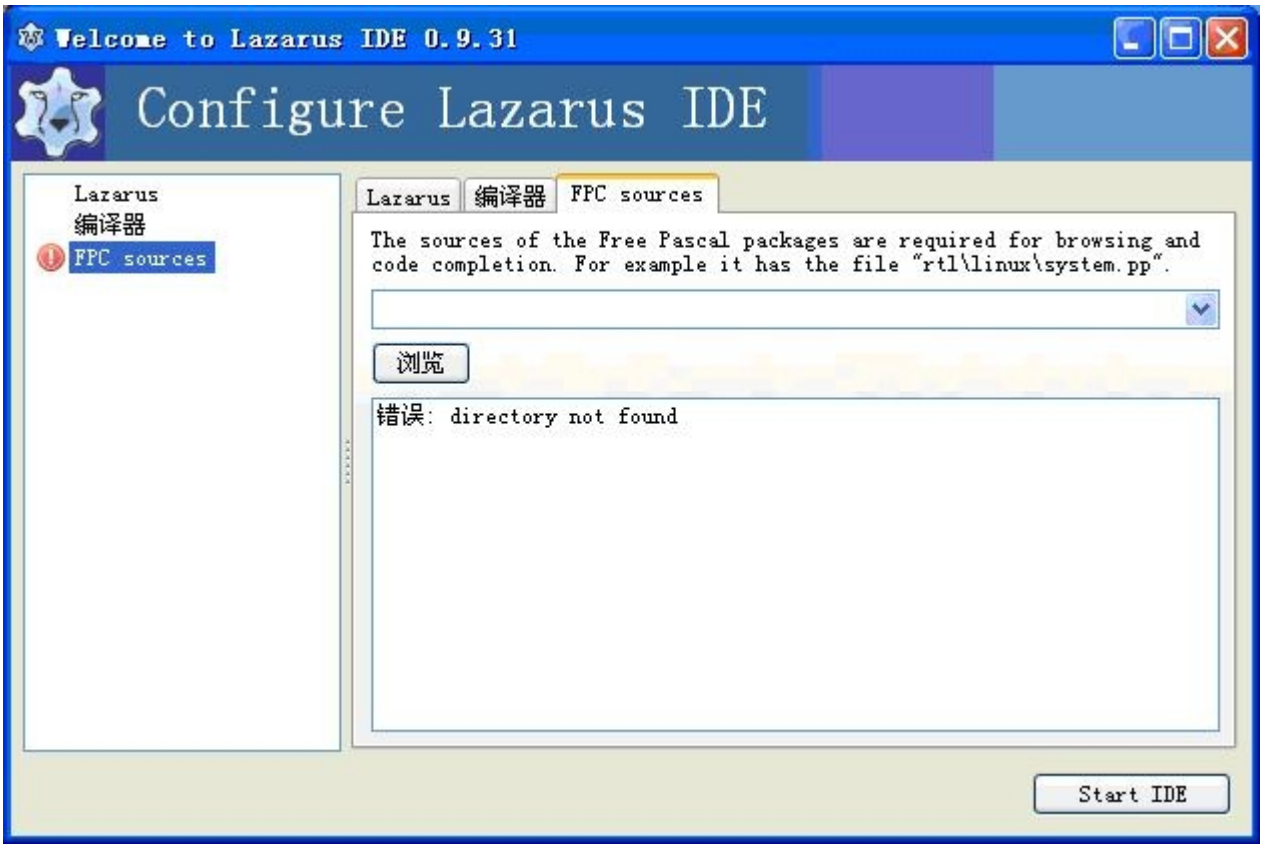
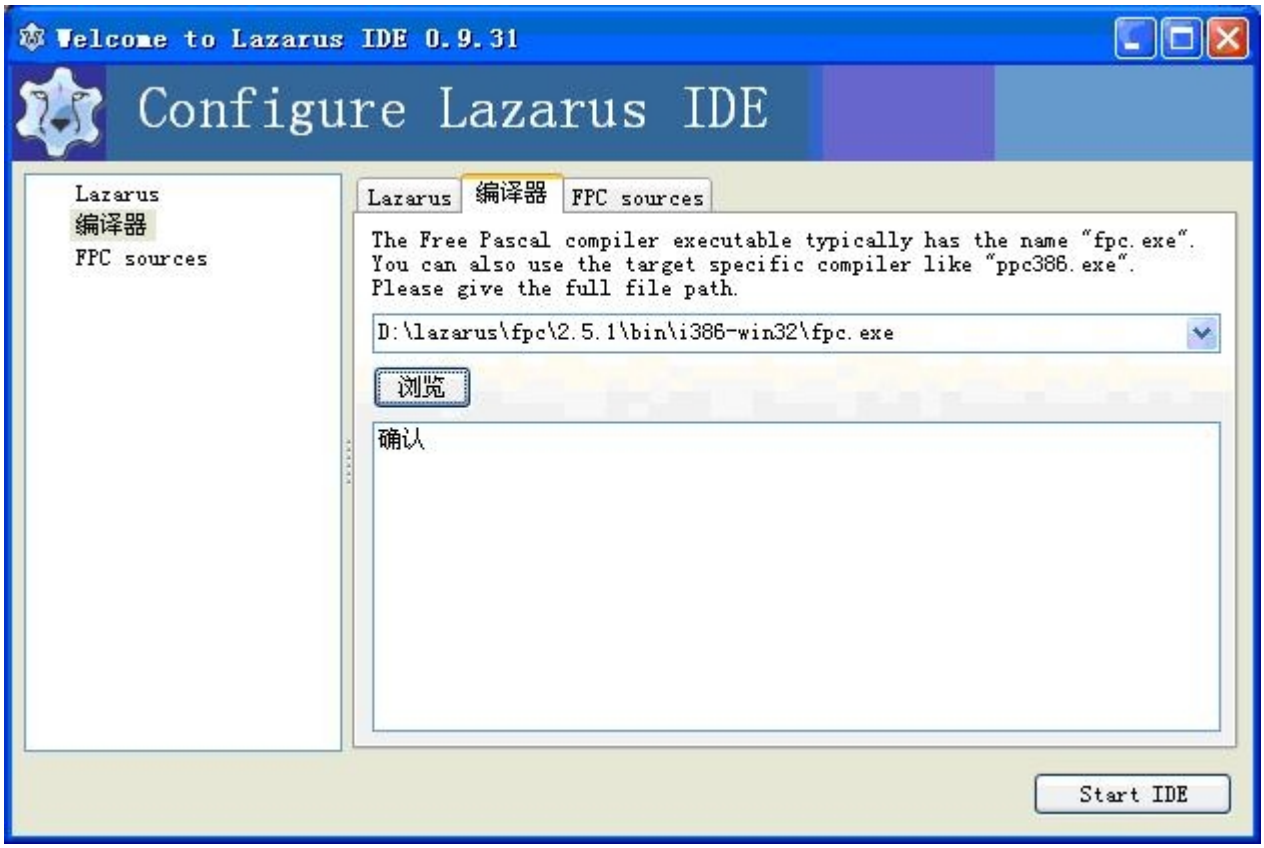


图 9

然后重新选择一下 fpc 的编译器及源码位置，



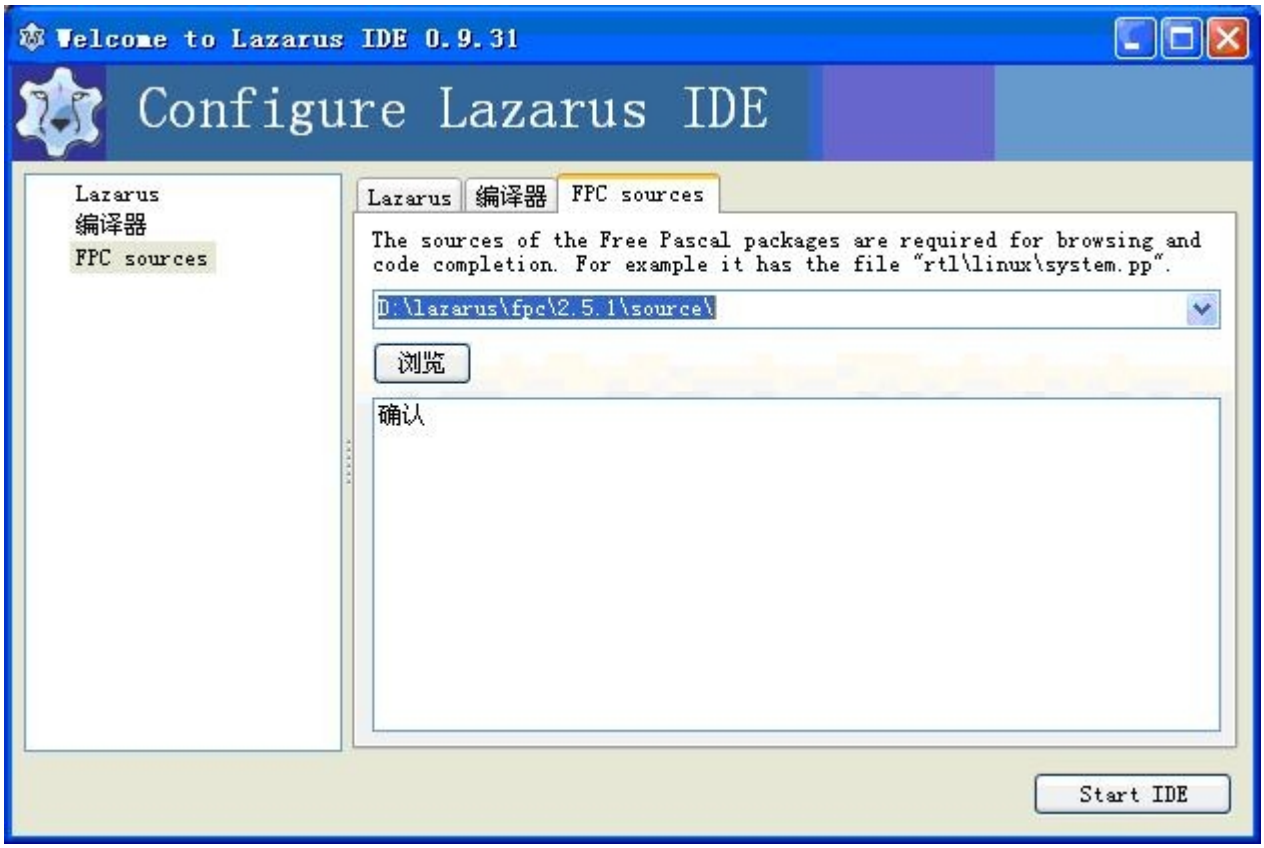


图 11

点 Start IDE，启动 lazarus，打开后选择工具->选项，查看环境是否正确设置：

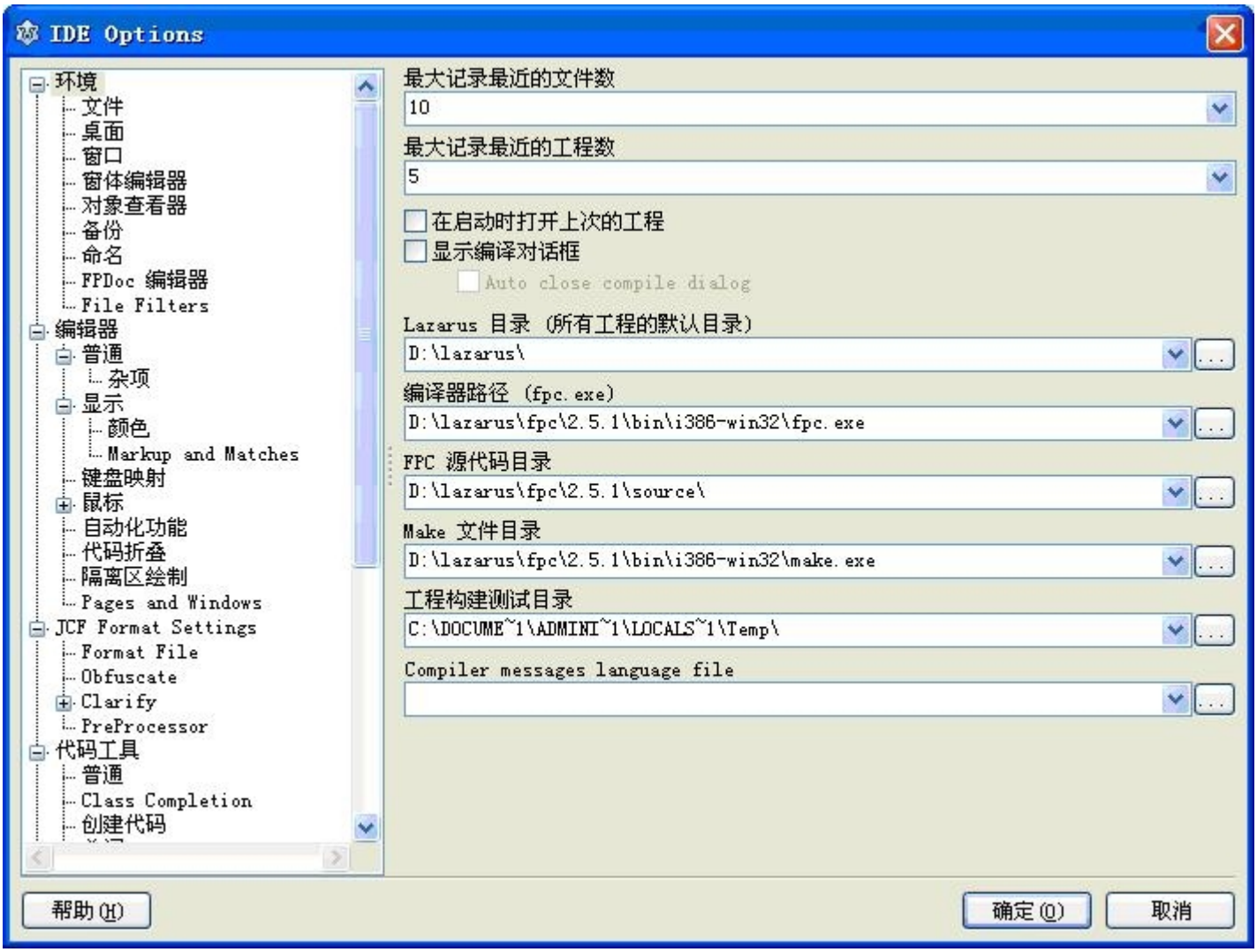


图 12

点工具->重新扫描 FPC 源代码目录。好了。注意，这样配置以后，只能编译生成 android 项目，其它标准工程会出错。

如果已经安装好 lazarus-0.9.31-35594M-fpc-2.5.1-20120229-win32.exe 这个安装包，请从这里开始。好了，可以打开 android 示例了，打开，然后进行工程设置。工程->工程选项->编译选项->路径：

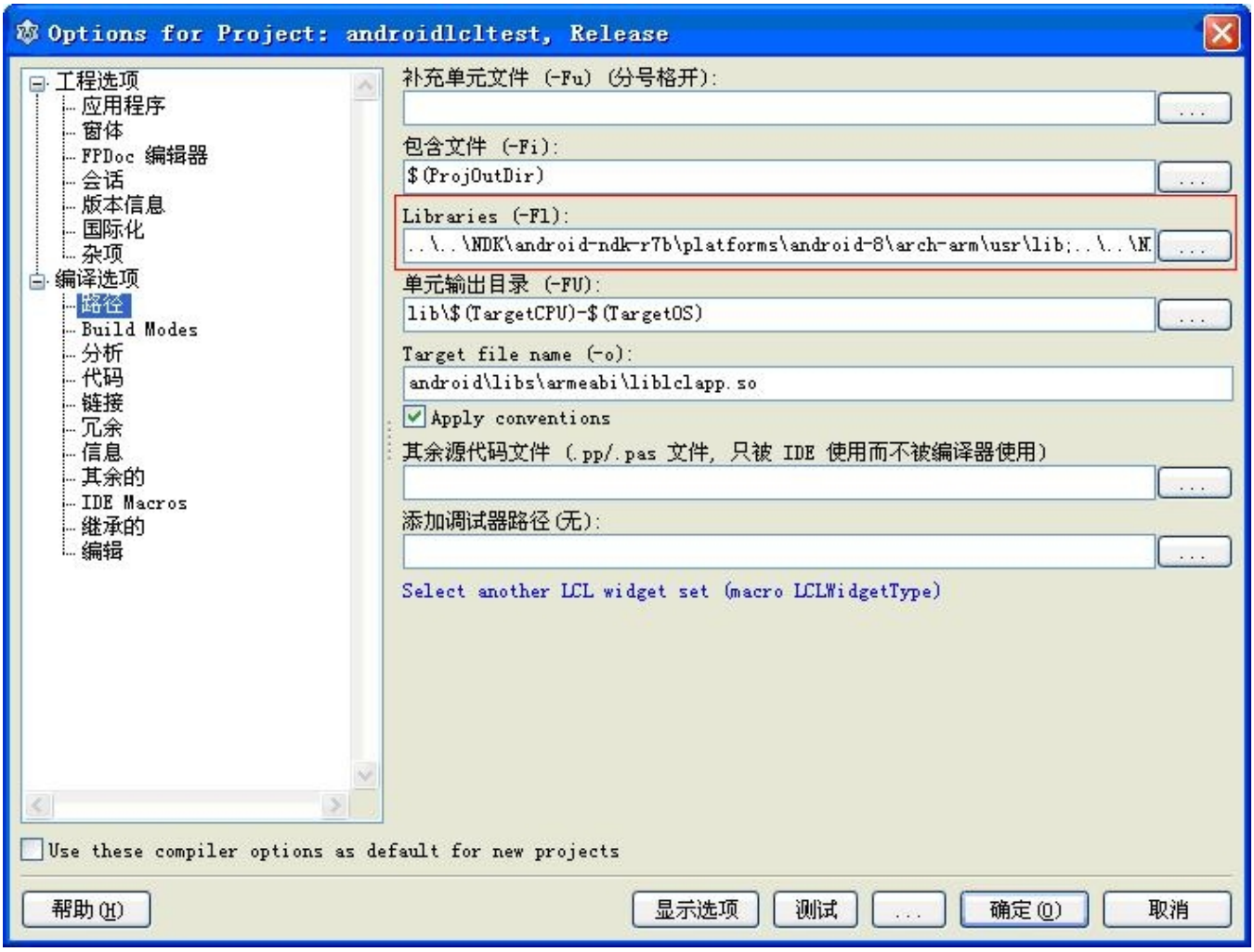


图 13

调整红色框部分，添加两个路径：

（1）：D:\AndroidFPC\NDK\android-ndk-r7b\platforms\android-8\arch-arm\usr\lib

（2）：

D:\AndroidFPC\NDK\android-ndk-r7b\toolchains\arm-linux-androideabi-4.4.3\prebuilt\windows\lib\gcc\arm-linux-androideabi\4.4.3

如图所示：

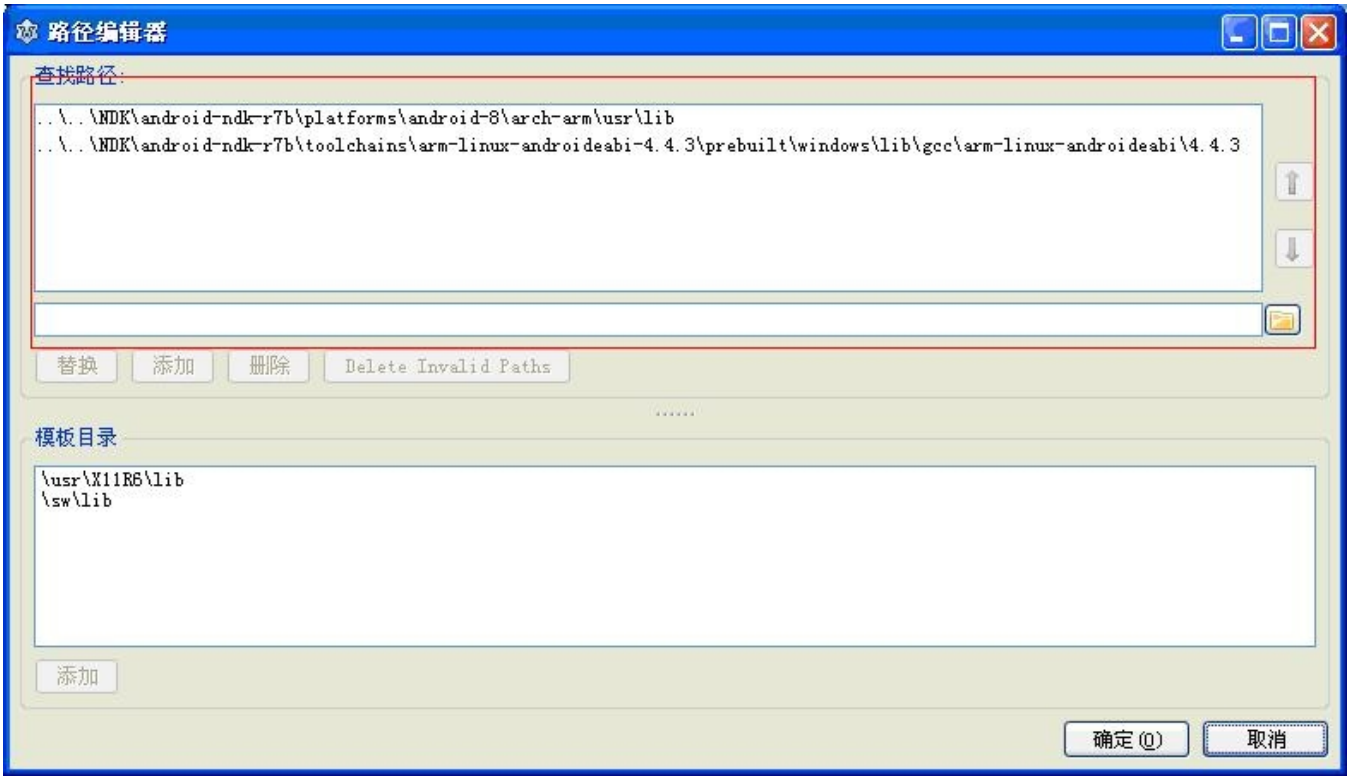


图 14

Build modes ：

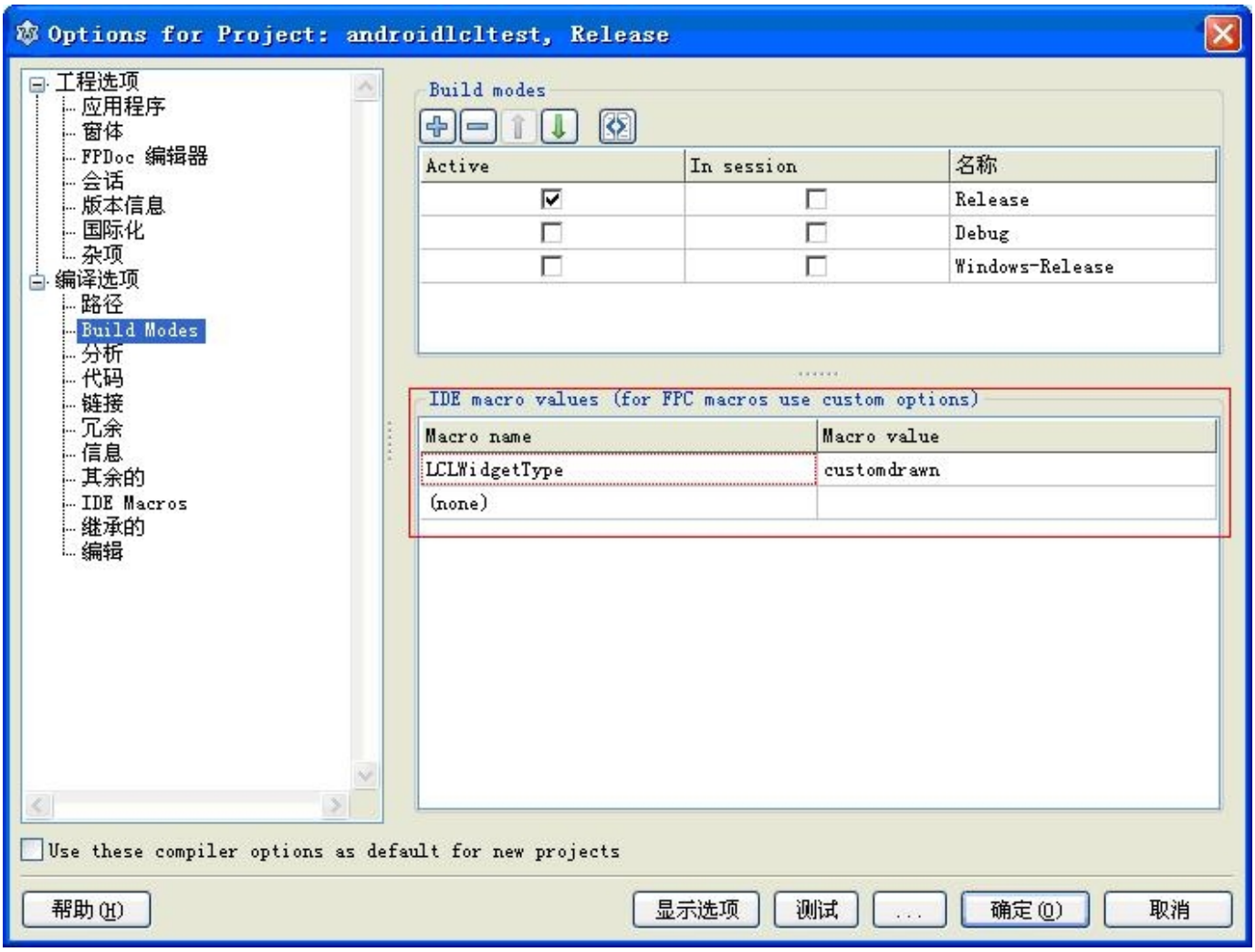


图 15

代码：

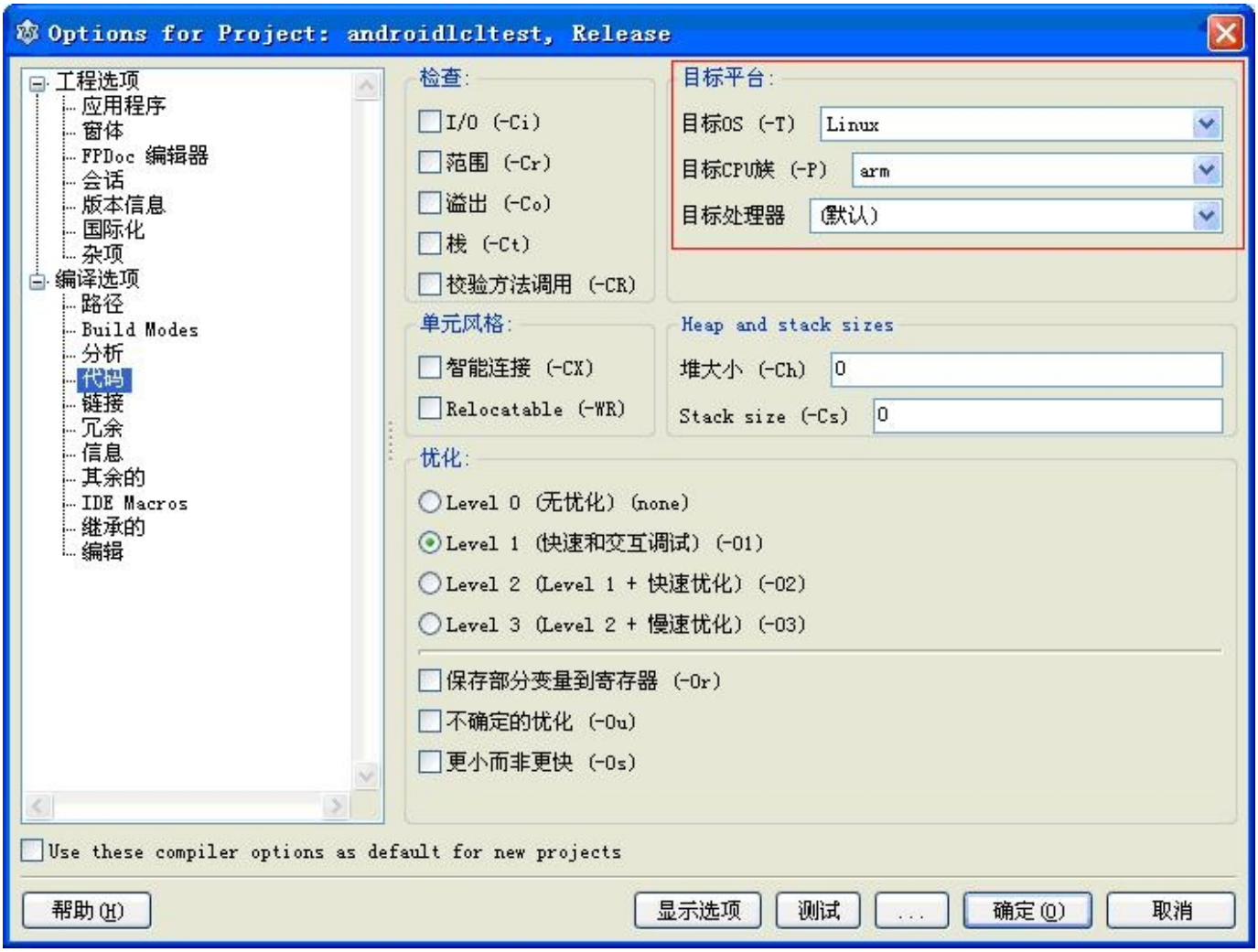


图 16

其它具体选项查看示例工程，不再一一列举。

2.2 开始编译及构建 android 示例。

如果已经按照以上的设置好了 android 工程，你现在可以 Shitf+F9 构建工程了。

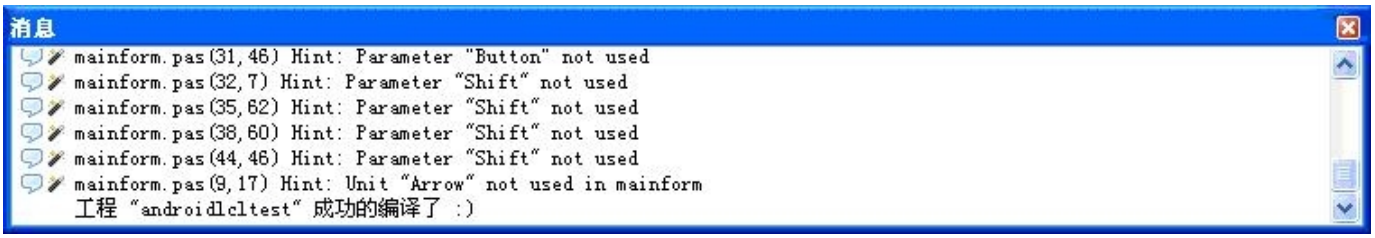


图 17

成功编译了，多么激动人心啊！

2.3 构建 APK。

在你使用命令行构建 APK 之前，请你先打开示例中 android 文件夹下的 local.properties 文件。打开后可以看到：

```
# This file is automatically generated by Android Tools.
# Do not modify this file -- YOUR CHANGES WILL BE ERASED!
#
# This file must *NOT* be checked in Version Control Systems,
# as it contains information specific to your local configuration.
# location of the SDK. This is only used by Ant
# For customization when using a Version Control System, please read the
# header note.
sdk.dir=/home/felipe/Programas/android-sdk-linux
```

修改最后一行为本机 Android SDK 的文件夹：sdk.dir=C:/Program Files/Android/android-sdk
进入命令行方式，开始构建 APK：
CD lazarus\example\androidlcl\android
Ant debug
我这里直接写了一个批处理，你们可以自己写一个。我这里就可以直接运行 demo 中的批处理文件：构建 APK.bat 。
好了，构建的 apk 文件在 lazarus\example\androidlcl\android\bin 中。**如果出现找不到 tools.jar，请把** C:\Program Files\Java\jdk1.7.0_03\lib\tools.jar **复制到** C:\Program Files\Java\jre7\lib **文件夹。**

2.4 安装 APK 到你的手机并且运行。

你可以通过 ADB 安装或者复制文件到你的手机的 SD 卡，就可以运行它。关于 ADB 请看 3.1 中的 ADB Install 的内容。在安装之前我们先创建 Android 模拟器，看 3.1 中的内容。

3．在 Android 模拟器和手机中运行 APK。

3.1 创建并设置模拟器。

运行程序中的 Android SDK Tools 的 android AVD 管理器，创建一个新的模拟器设备：

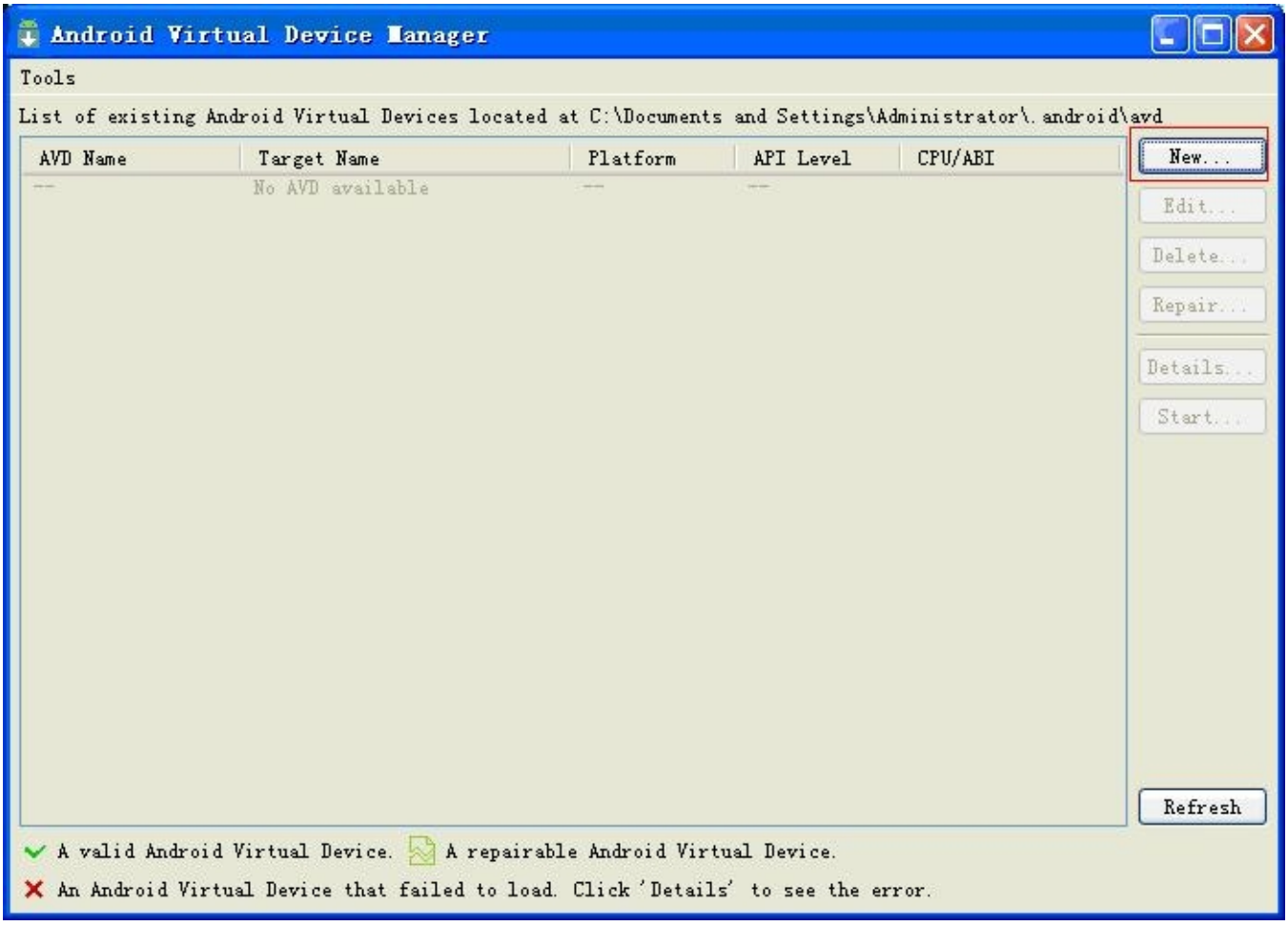


图 18

点击 New，设置如下：

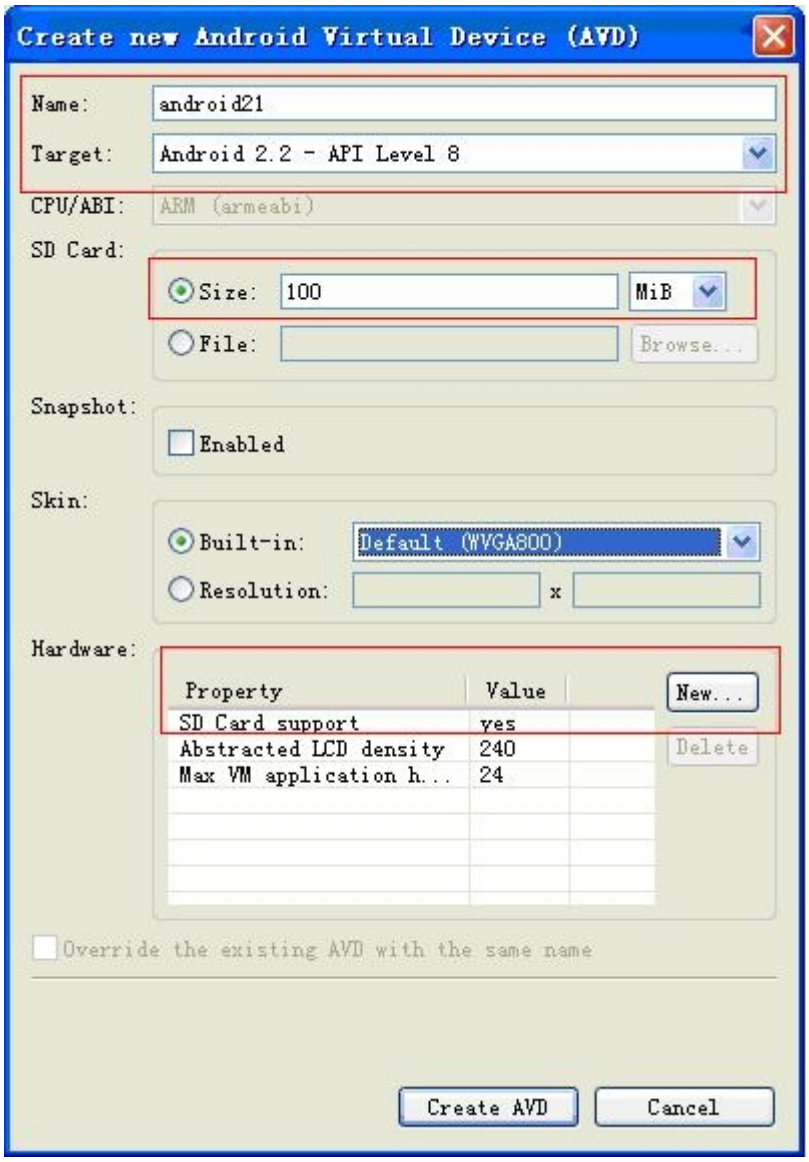


图 19

SD Cardsupport 需要点右边的 New 才能添加进去。

点击 Create AVD，好了，创建了一个模拟器了。

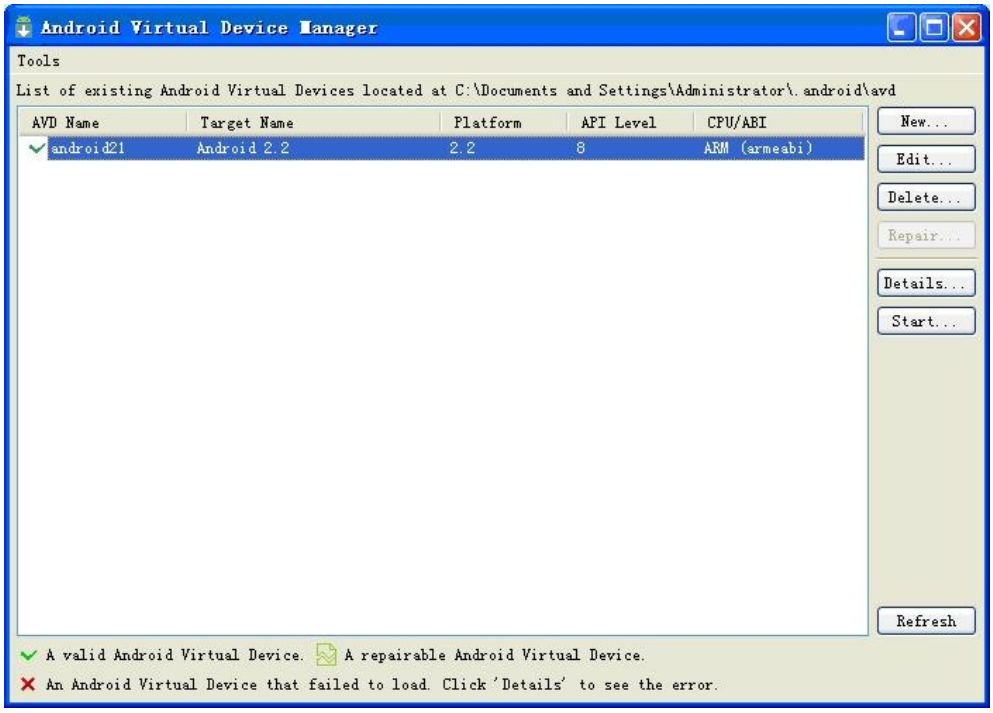


图 20

3.2 运行模拟器

现在我们运行模拟器。点击 Start。

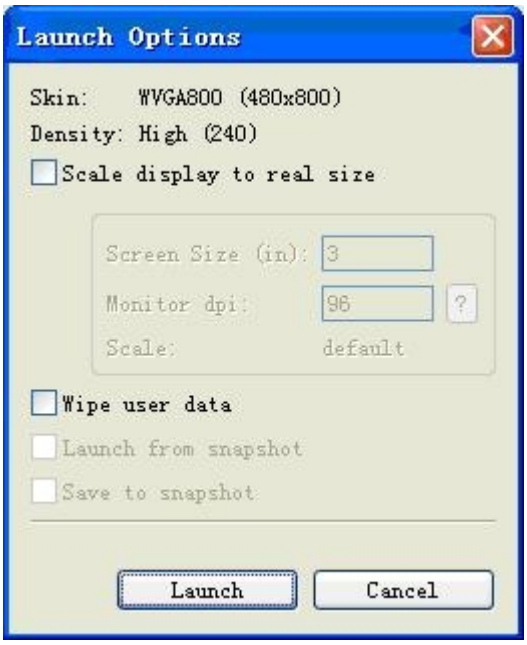


图 21

点 Launch，开始运行了，等待几分钟，模拟启动完成。

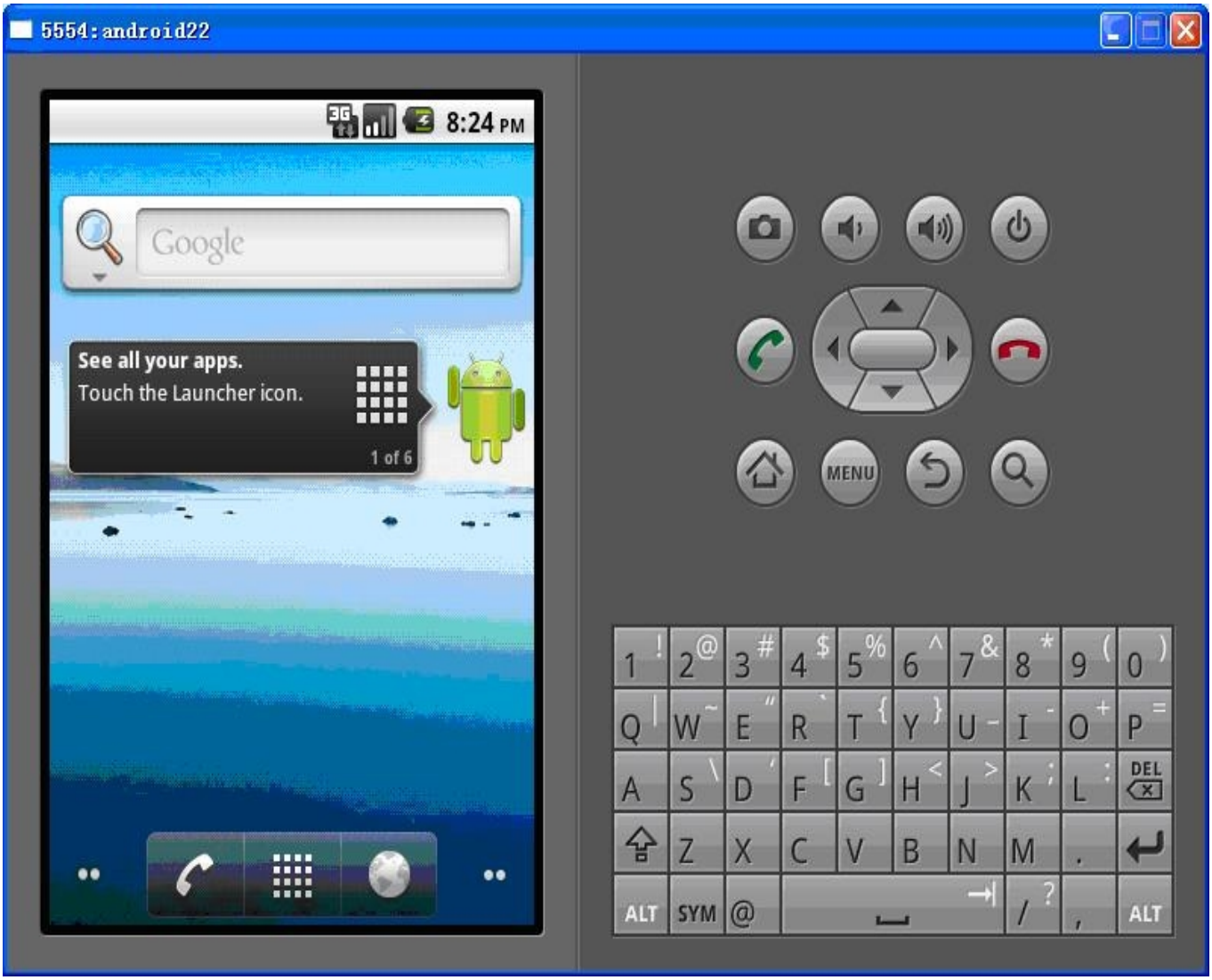


图 22

如果第一次出现运行错误，不要管它，请关闭，重新再开始。

3.3 安装和运行 APK 程序

启动完成模拟器才可以安装程序。

进入命令行方式，进入文件夹：C:\Program Files\Android\android-sdk\platform-tools

使用如下命令把前面生成的 apk 文件安装进去。

Adb install <apk 文件路径>

我这里是 adb install D:\AndroidFPC\demo\demo\android\bin\LCLExample-debug.apk，或者运行 demo 中的批处理文件：安装 APK.bat，等它完成。



图 23

显示 Success 就成功应用软件安装进模拟器了。

Adb 常用命令：

Adb install <apk 文件路径>　　安装 apk 到模拟器

Adb uninstall<apk 文件路径> 卸载 apk

Adb devices　　列出所有模拟器设备

Adb logcat　　调试器实时记录

看我们的 LCL Test：



图 24

运行后：

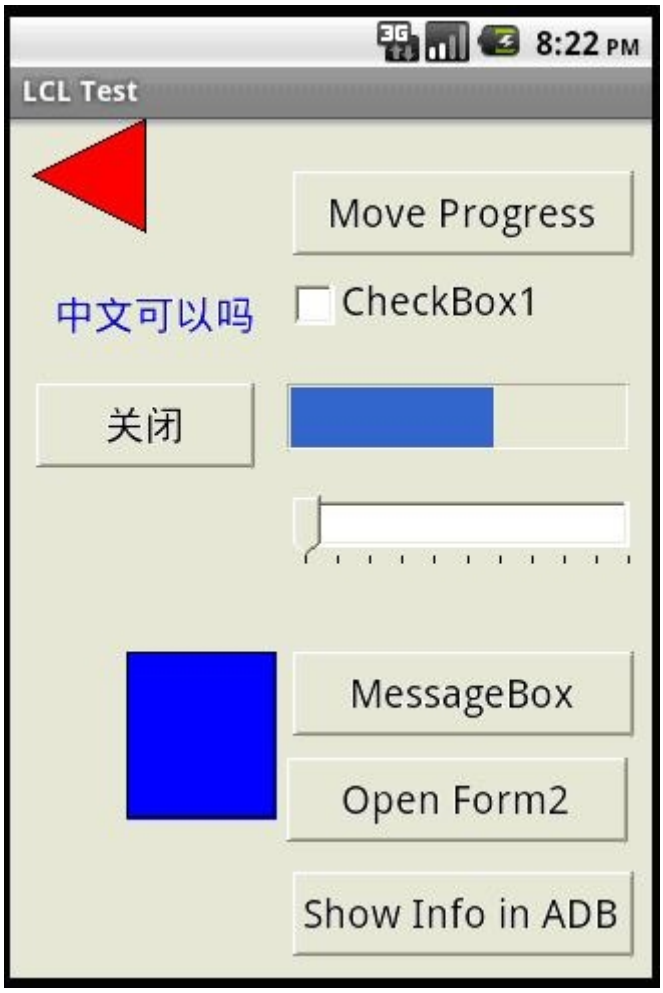


图 25

4．示例程序 NDK OpenGL

引用地址：http://wiki.freepascal.org/Android_Programming

这些需要用到 2.3 的 SDK，请先下载安装好。

4．1 下载安装 Android NDK ，Android SDK 和 Ant

4．2 安装 FPC 交叉编译器。

4．3 SVN 下载最新的 lazarus-ccr sourceforge 源码：

https://lazarus-ccr.svn.sourceforge.net/svnroot/lazarus-ccr

或者直接下载：lazarus-ccr/bindings/android-ndk 里面的源码就可以了。

4．4 打开 lazarus-ccr/bindings/android-ndk/examples/opengltest/opengltest.lpi ，路径设置一下，具体看示例。编译生成。

4．5 配置一下 local.properties 文件。

4．6 构建 APK，安装模拟器 2.3.3。

5．如何创建一个新的 LCL Android 应用程序

引用地址：

http://wiki.freepascal.org/Custom_Drawn_Interface/Android#How_to_create_an_LCL_Android_Application

以下部分是参考官网和自己研究所得：

创建一个新的 LCL-CustomDrawn-Android 应用程序可以简单的从示例程序复制。这个示例程序可以从 lazarus/examples/androidlcl 中找到。然后你需要修改构建文件来变成你自己的新的工程名称和新的 Java 包名称。

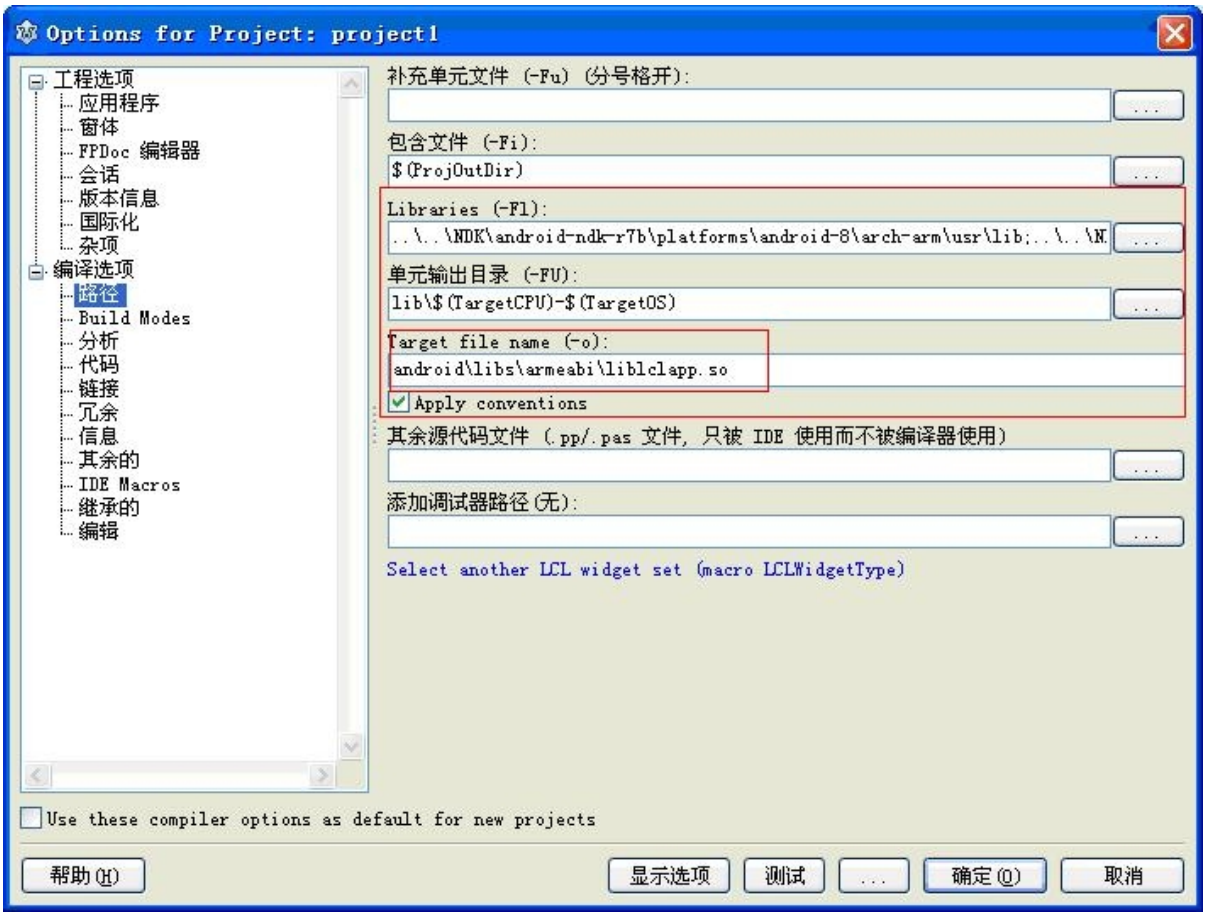
5.1 创建 LPI 文件

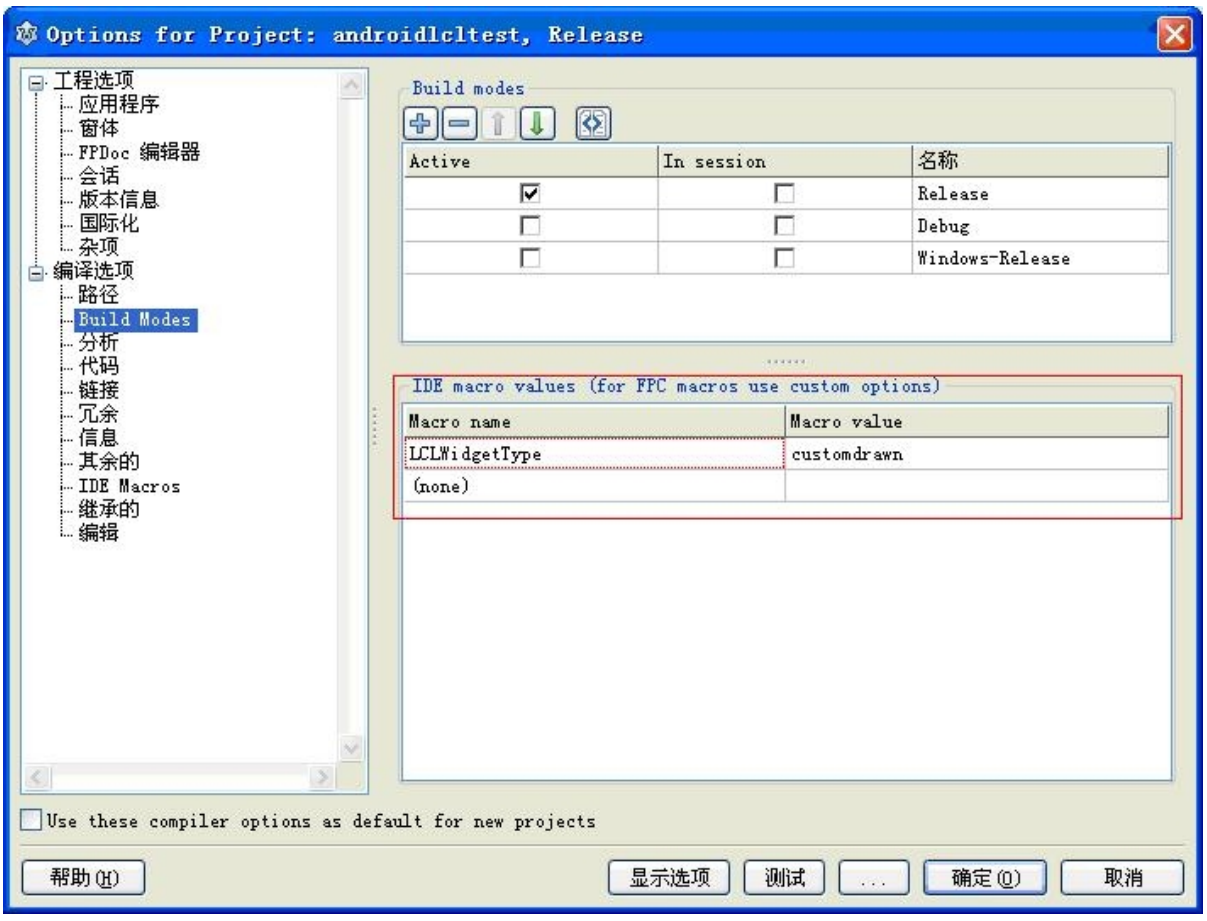
此时你需要一个单独的 LPI 文件来用于安卓应用程序版本，但是所有其它代码可以被共享。

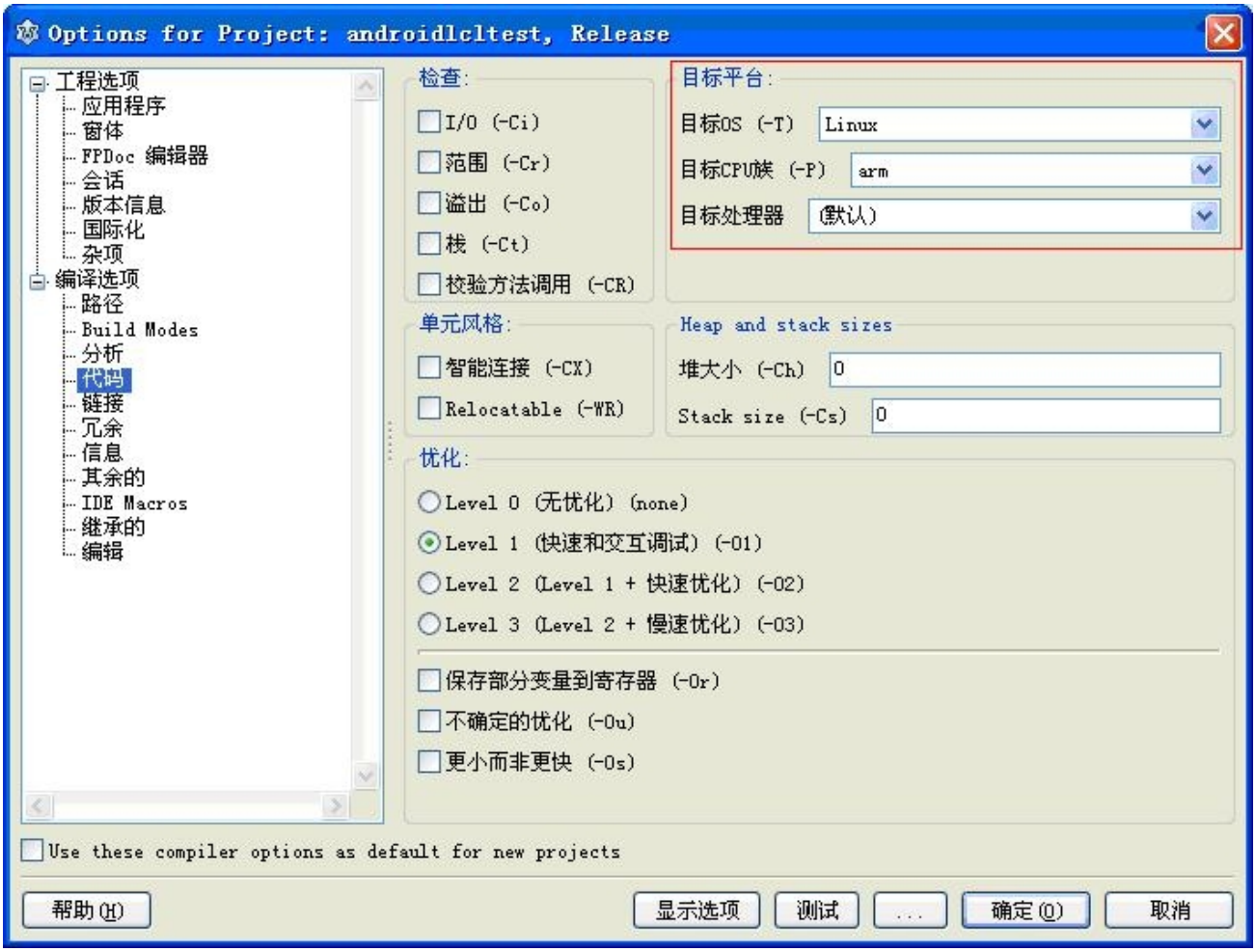
创建它可以从新建->工程->库来建立一个新的文件，然后从示例程序中来改写其它的代码。你需要改写工程 LPR 文件 exported 部分的 JNI 属性名字为你的安卓包的名字。

5.2 构建库文件

首先，工程去掉 debug 信息。Debug 信息对安卓没用，并且它使得执行文件的体积太大。看图：






Options for Project: androidlcltest, Release
工程选项
应用程序
窗体
FPDoc 编辑器
会话
版本信息
国际化
杂项
编译选项
路径
Build Modes
分析
代码
链接
冗余
信息
其余的
IDE Macros
继承的
编辑
检查:
I/O (-Ci)
范围 (-Cr)
溢出 (-Co)
栈 (-Ct)
校验方法调用 (-CR)
目标平台:
目标OS (-T) Linux
目标CPU族 (-P) arm
目标处理器 (默认)
单元风格:
智能连接 (-CX)
Relocatable (-WR)
Heap and stack sizes
堆大小 (-Ch) 0
Stack size (-Cs) 0
优化:
Level 0 (无优化) (none)
Level 1 (快速和交互调试) (-O1)
Level 2 (Level 1 + 快速优化) (-O2)
Level 3 (Level 2 + 慢速优化) (-O3)
保存部分变量到寄存器 (-Or)
不确定的优化 (-Ou)
更小而非更快 (-Os)
Use these compiler options as default for new projects
帮助(H)
显示选项
测试
...
确定(O)
取消

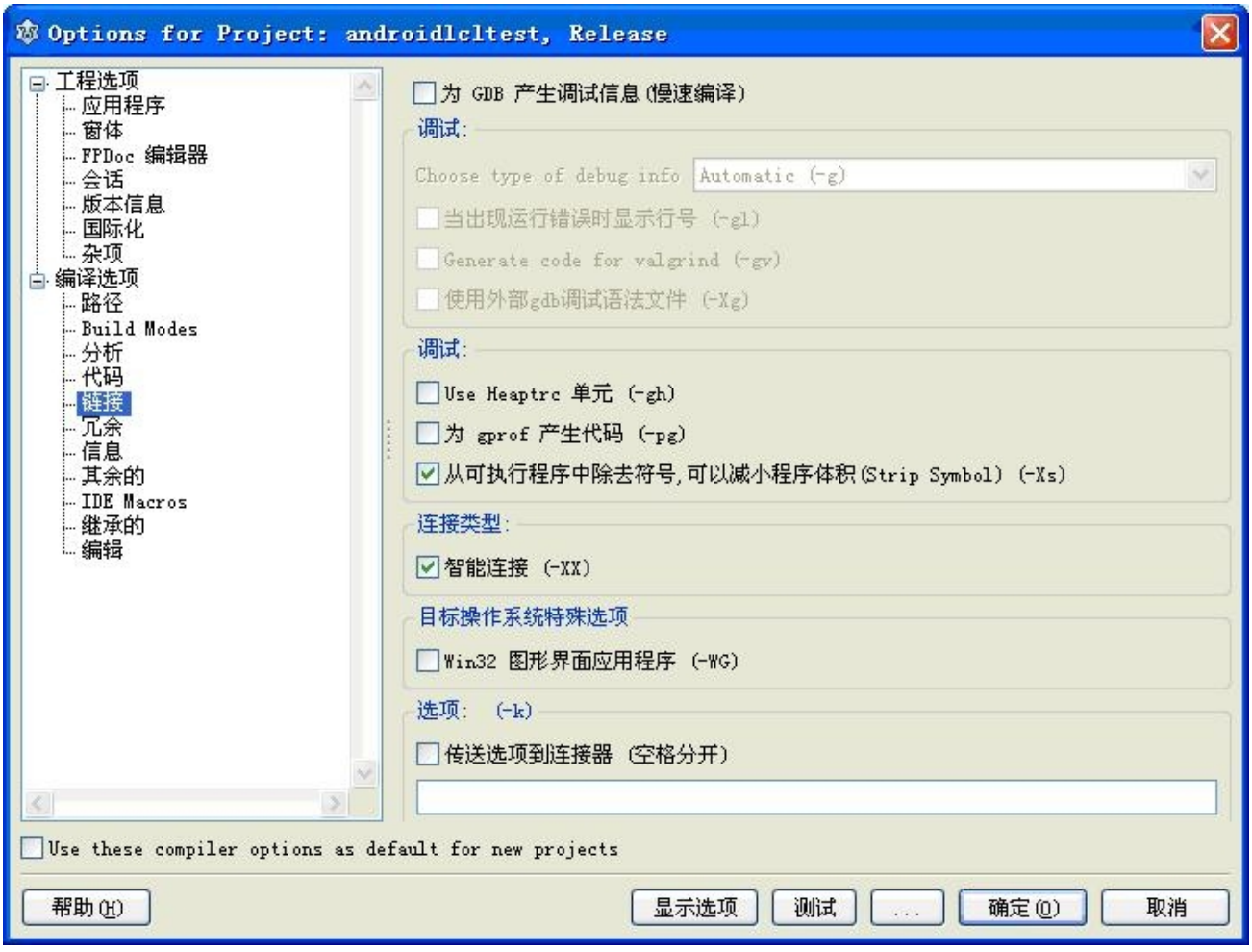

Options for Project: androidlcltest, Release
工程选项
应用程序
窗体
FPDoc 编辑器
会话
版本信息
国际化
杂项
编译选项
路径
Build Modes
分析
代码
链接
冗余
信息
其余的
IDE Macros
继承的
编辑
为 GDB 产生调试信息(慢速编译)
调试:
Choose type of debug info Automatic (-g)
当出现运行错误时显示行号 (-gl)
Generate code for valgrind (-gv)
使用外部gdb调试语法文件 (-Xg)
调试:
Use Heaptrc 单元 (-gh)
为 gprof 产生代码 (-pg)
从可执行程序中除去符号,可以减小程序体积(Strip Symbol) (-Xs)
连接类型:
智能连接 (-XX)
目标操作系统特殊选项
Win32 图形界面应用程序 (-WG)
选项: (-k)
传送选项到连接器 (空格分开)
Use these compiler options as default for new projects
帮助(H)
显示选项
测试
...
确定(O)
取消

图 29

5.3 创建安卓工程结构

和一般的 LCL 工程构建比起来，像这样配置一个正确的 LPI 并且拥有一个库主工程文件，也只需要添加所有的安卓工程结构。完全从示例程序复制就可以了，然后修改适应新的工程。然后更改 SDK 路径，包名称。包的名称需要在新的工程文件中修改。在文件夹 android/src/包名称，在 activity java 源代码并且在工程的库文件 pascal 代码中，在 exports 部分。

上面是翻译了官网和，

自己研究如下：包文件名称和路径要和 LPR 中的文件名要一样。这里的包指的是 Android 包。

```
library andtest;

{$mode objfpc}{$H+}

uses
  customdrawnint,
  Interfaces,
  Forms, customdrawn,
  Unit1;
exports
  Java_com_pascal_lclproject_LCLActivity_LCLOnTouch name 'Java_com_pascal_truetest_LCLActivity_LCLOnTouch',
  Java_com_pascal_lclproject_LCLActivity_LCLDrawToBitmap name 'Java_com_pascal_truetest_LCLActivity_LCLDrawToBitma
  Java_com_pascal_lclproject_LCLActivity_LCLOnCreate name 'Java_com_pascal_truetest_LCLActivity_LCLOnCreate',
  Java_com_pascal_lclproject_LCLActivity_LCLOnMessageBoxFinished name 'Java_com_pascal_truetest_LCLActivity_LCLOnM
  Java_com_pascal_lclproject_LCLActivity_LCLOnKey name 'Java_com_pascal_truetest_LCLActivity_LCLOnKey',
  Java_com_pascal_lclproject_LCLActivity_LCLOnTimer name 'Java_com_pascal_truetest_LCLActivity_LCLOnTimer',
  Java_com_pascal_lclproject_LCLActivity_LCLOnConfigurationChanged name 'Java_com_pascal_truetest_LCLActivity_LCLC
  Java_com_pascal_lclproject_LCLActivity_LCLOnSensorChanged name 'Java_com_pascal_truetest_LCLActivity_LCLOnSensor
```

```
name 'Java_com_pascal_truetest_LCLActivity_LCLOnTouch',
tmap name 'Java_com_pascal_truetest_LCLActivity_LCLDrawToBitma
 name 'Java_com_pascal_truetest_LCLActivity_LCLOnCreate',
eBoxFinished name 'Java_com_pascal_truetest_LCLActivity_LCLOn
me 'Java_com_pascal_truetest_LCLActivity_LCLOnKey',
name 'Java_com_pascal_truetest_LCLActivity_LCLOnTimer',
urationChanged name 'Java_com_pascal_truetest_LCLActivity_LCL
Changed name 'Java_com_pascal_truetest_LCLActivity_LCLOnSenso
```

图 31

以上两图是同一张图，看红色框部分。再看 Android 的 java 包的文件路径，看下图。



图 32

看文件路径名称同样为：src\com\pascal\truetest\LCLactivity。

注意：Java 文件要加入到工程中去，打开 java 文件然后菜单 工程->添加代码编辑器中的文件到工程。后来研究了一下，这里不一定。

还要调整编译后的 so 文件名称和 java 中的一样。说明如下：

LCLActivity.java 文件中最后部分，第 621 行：

```
616    static
617    {
618      try
619      {
620        Log.i("lclapp", "Trying to load liblclapp.so");
621        System.loadLibrary("lclapp");
622      }
623      catch(UnsatisfiedLinkError ule)
624      {
625        Log.e("lclapp", "WARNING: Could not load liblclapp.so");
626        ule.printStackTrace();
627      }
628    }
629  }
630
```

图 33

看黑色框中的 lclapp　　，那么你的编译后的导出库名称也要 lclapp ，看工程选->编译选项->路径->Target file name，看图：



图 34

注意，lclapp.so 前的 lib 和 so 文件存放路径位置。

安卓程序的图标和文件名称的更改：

图标和文件名称存放在以下位置中 android\res\中 ，drawable-hdpi、drawable-mdpi、drawable-ldpi 三个文件夹，分别存放 72x72、48x48、36x36 大小两种图标。而 values 文件中的 strings.xml 文件中就是存放安卓文件名称了，打开修改　　<string name="app_name">LCLTest</string>部分的 LCL Test 就可以了。如是是中文名称，请把文件另存为 utf8 格式。

生成的 APK 名称在这里定义：android\build.xml 中定义，在 2 行：

<projectname="LCLExample" default="help">　　把 LCLExample 改为你自己的名称。

6．在 Android2.2(API 8)中可用的 NDK 库

LCL-CustomDrawn 设计目标为 Android2.2+ 并且 API 库在 NDK 中如下支持：

(1) libc.so
(2) libdl.so(linker)
(3) OpenGL ES 1
(4) OpenGL ES 2
(5) libjnigraphics.so
(6) liblog.so
(7) libm.so
(8) libthread_db.so
(9) libstdc++.so

当前 LCL-CustomDraw-Android 使用 libjnigraphics.so 和 liblog.so 。

7．本地文本绘制

在安卓中本地文本绘制有一些费解并且这个帮助文件也相当糟糕。一些已经了解的在如下：

(1)绘制文本开始以 Y 坐标作为基准线，

(2)没有帮助文件描述 Paint.setTextSize 中的尺寸是如何的。有些人说是像素，但是我看到一些不同，它好像依赖于不同设备之间。

(3)也有一些好的事情：

- 可以绘制透明的位图并且使用 alphi 透明度去实现 anti-aliasing(反钜齿处理)和 subpixel rendering(子像素渲染)。文本变得十分漂亮。
- 也可以绘制 ARGB 32 位非像素格式也是相当漂亮的。

8．Lazarus 开发组的实现路线

http://wiki.freepascal.org/Custom_Drawn_Interface/Android#Roadmap 2012-2-24

(1) 构建 Lazarus Custom Drawn 控件组。已完成
(2) 为安卓 API 初始化绑定。已完成
(3) 创建一个应用程序自动生成绑定。已完成
(4) 开始一个新的 widgetset（组件集，构件集）。已完成
(5) 实现对 JNI 的支持。已完成
(6) 合并 Lazarus Custom Drawn 控件到 LCL 并且用它们实现所有最基本的控件。已完成
(7) 为安卓添加文本支持。已完成
(8) 实现基本窗口客户区滚动。已完成
(9) 为桌面平台添加手机模拟器。进行中
(10) 改善手机滚动。尚未进行

(11) 在 X11 中添加文本支持。尚未进行
(12) 在 LCL 中添加 DPI 感应度和自适应。尚未进行
(13) 添加非本地化文本支持通过 pasfreetype 。尚未进行

9. 关于 LazarusCustom Drawn 接口

一个跨平台 LCL 后端，完全在 lazarus 内使用 Object Pascal 编写的。Lazarus 到 Android 的接口。LCL-CustomDrawn-Android 有以下特性：
此部分尚未完成整理。

10. 官方论坛看到的资料

1 引用页：http://www.lazarus.freepascal.org/index.php/topic,15951.0.html。Android NDK 目前为止所有的转换 pas 都在 lazarus/lcl/interfaces/customdrawn/android 中。但是这个转换目前不包括摄像头和手机通讯簿部分。转换的版本是 Android2.2+。对 SDK 库，非常巨大，可以通过 JNI 直接访问它们。有人提到如何设置背景为黑色，他说在 common_draw 中修改，我想是在 customdrawn_common 单元，这个要测试一下。

2 月 29 日版：lazarus-0.9.31-35594M-fpc-2.5.1-20120229-win32.exe 安装包 115 网盘：
http://115.com/file/belt6tb7
3 月 4 日新版：lazarus-0.9.31-35701M-fpc-2.5.1-20120304-win32.exe 安装包 115 网盘：
http://115.com/file/e7scut2m
3 月 15 日新版：lazarus-0.9.31-36024M-fpc-2.5.1-20120315-win32.exe 安装包 115 网盘：
http://115.com/file/e7qlao2q